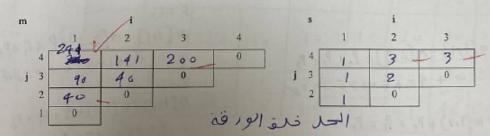
Compare dynamic programming and standard recursion by filling out the table below.

Algorithm	Top-down or bottom- up?	Solve the same subproblem once?	Always solve all sub- problems		
Dynamic Programming	60t to MUPV	905/	yes /		
Standard Recursion	bottomup	yes /	yes/		

Problem 2 (6 points)

Let  $A_1, \ldots, A_4$  be matrices with dimensions  $5 \times 2$ ,  $2 \times 4$ ,  $4 \times 5$ ,  $5 \times 10$ , respectively. In finding an optimal parenthesization of the matrix chain product  $A_1 * A_2 * A_3 * A_4$ , we use two tables  $m[\cdot, j]$  and  $s[\cdot, j]$  below. Here m[i,j] stores the optimal cost of computing subchain  $A_i ... A_j$  and s[i,j] records the index k where the optimal parenthesization splits  $A_i ... A_j$  between  $A_k$  and  $A_k + I$  for some k with  $i \le k \le j-1$ .

b- Fill the empty entries in the two tables. Show your work in each case.



c- Now, give the optimal parenthesization of the matrix chain product A1\*A2\*A3\*A4. Show how you came up with the solution using the tables above.

(A) (A2 A3) (A4))

explain: using the table(s)above \$c(ecte & the place of the parentasis for example by using \$[1,6] imadage & to know that i have to parenthesiz a feer Al isolating it from the rest.

Problem 3 (6 points)

Solve the following instance of the Knapsack Problem using Dynamic programming paradigm. The maximum allowed weight is Wmax = 10.

i	1	2	3	4
Vi	20	10	15	31
Wi	6	1	2	5

a- Give the recursive equation you used to define the data structure needed for your dynamic programming solution. Then, fill the proposed data structure.

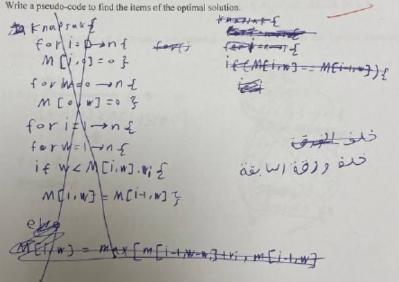
	1		- L	2	3	4	9	6	7	8	9	10
	0	0	0	0	0	10	0	10	10	10	10	101
V i = 20 W i = 6	1	0	0	0	0	6	0	20	20	20	20	
V ( = 1 0 W ( = 1	2	0	10	10	10	10	10	20	30	30	30	30
v i= 15 w i= 2	3	0	10	15	25	25	25	25	30	35	45	45
Vi= 31 Wi= 5	4 L	0	10	15	25	25	31	41	46	56	50	56
MELLAND LINE		m [i-l,	7-W3+	1 house				1				1

Knap sake WHILE let R= n, let solution [], w let D= w while (R & i) L AND if (m([R, w] = = m [P-1, w]) R = P-1; } else g Solution(1) = n[R, w] n[R,D] = m[R-1, D-Wi] 3 b- What is your solution to this instance of the Knapsack problem?

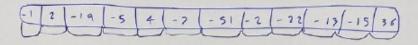
$$(4,3,2) \qquad 4 \qquad w_4 + w_9 + w_7 = 5 + 2 + 1 = 8$$

$$v_4 + v_3 + v_2 = 3(+15 + 10 = 56)$$

c- Write a pseudo-code to find the items of the optimal solution.







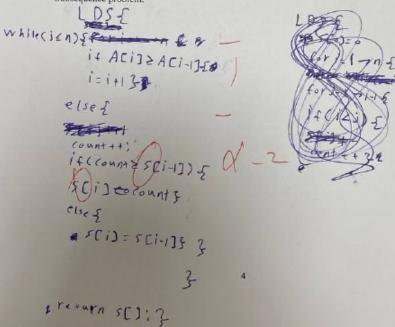
Problem 4 (5 points)

The Longest Decreasing Subsequence problem is defined as follows: Given a sequence of n real numbers A[1]...A[n], determine a subsequence (not necessarily contiguous) of maximum length in which the values in the subsequence form a strictly decreasing sequence.

## Example:

The length of the longest decreasing Subsequence in [-1, 2, -19, -5, 4, -7, -51, -2, -22, -13, -15, 36] is 5.

a- Give the pseudo-code of a Dynamic programming algorithm that solves the Longest Decreasing Subsequence problem.



b- What is the time complexity of your algorithm? Prove it!

T(n) = 6n + 1

 $6n+1 \leq 7n$ 

(=7

No=1